Aquecimento 02 - Projeto

Casa de Cambio via Terminal

Sua aplicação já está sendo utilizada, mas agora você está passando por um review técnico de outro desenvolvedore. ELe traz alguns pontos relevantes para você melhorar no seu código:

- Orientação a Objetos
    - Criar uma classe `Cashier` que registre o início da operação do caixa, com as informações de cotação e quantidade de moedas disponíveis;
    - Criar na classe `Cashier` métodos relativos às operações principais do sistema (compra e venda de cada moeda, exibição do status do caixa);
    - Atualizar o arquivo principal da aplicação para instanciar um `cashier` no início da execução e usar seus métodos de acordo com a opção escolhida no menu;
- Ruby Way

  Além de organizar melhor seu código com a classe `Cashier`, o desenvolvedor sugeriu alguns ajustes relativos ao jeito Ruby de se programar:
    - Use constantes para definir as opções do menu e facilitar a leitura do código por outros devs;
    - Evite concatenar strings, use interpolação sempre que precisar manipular/misturar dados gerando uma string;
    - Não use `return` no final de métodos. Em Ruby a última instrução executada é retornada automaticamente em todos os métodos;

Banco de Dados

Falamos da importancia de bancos de dados em nossa live/vídeo e fizemos uma pequena demonstração. Agora chegou a hora de salvar nossas transações de cambio em uma tabela usando SQLite.

SQLite é um banco de dados relacional simples, baseado em arquivo e que é usado como padrão em aplicações Rails para agilizar o início do desenvolvimento.

Para nossa aplicação devemos:

- Criar um banco de dados chamado: 'cambio.db'
- Criar uma tabela 'transactions', com colunas adequadas para armazenar os dados das transações
    - o nome das colunas deve refletir os atributos da sua classe;
    - o tipo de cada coluna fica a seu critério;
    - use AUTOINCREMENT na coluna ID das transações;
- Ao salvar uma transação, devemos inserir os dados na tabela 'transactions';
- Ao consultar as transações do dia, fazer uma consulta no banco de dados;

Referências

Introdução a SQL: https://www.w3schools.com/sql/sql_intro.asp

Tutorial para SQLite3: https://www.tutorialspoint.com/sqlite/index.htm

Gem SQLite3: https://github.com/sparklemotion/sqlite3-ruby

Bônus

- Crie uma tabela no banco de dados para armazenar o `cashier`;
- Ao criar um caixa, insiera na tabela `cashiers` informando o dia/mes/ano da ação do caixa;
- Cada vez que o programa for executado, verifique se já existe um registro na tabela `cashier` para o dia atual
    - caso sim, exiba a cotacao e as quantidades de de moeda;
    - pergunte se deseja atualizar essas informações, caso sim faça novamente as pergiuntas e execute um `UPDATE` no registro;
    - caso não exista o caixa, crie um novo;

Bônus Hard

- Altere o fluxo de inicialização e a classe `Cashier` para receber o nome do operador do caixa;
- Registre o nome do operador na tabela `cashiers`
- Na tabela transaction, crie uma chave estrangeira (foreign key) para a tabela `cashiers`, relacionando as transações com um caixa;
- Ao solicitar o histórico do dia, busque somente transações do caixa corrente;